CLAIMS

What is claimed is:

1. A method of reducing a size of a netlist for a target architecture, comprising:

create a netlist of objects for the target architecture;

identify objects specific to the target architecture that are repeated regularly to identify potential dummy objects;

create a list of objects used by a predetermined design in the target architecture;

form a list of unused objects in the target architecture from the netlist of objects and the list of objects used by the predetermined design;

replace at least one object in the list of unused objects with an appropriate dummy object to form a modified netlist; and

simulate the modified netlist.

2. The method of claim 1, wherein the step of forming the list of unused objects comprises the steps of subtracting the list of objects used by the predetermined design from the netlist of objects.

3. The method of claim 1, wherein the step of replacing at least one object comprises the step of replacing objects in the list of unused objects that are repeated regularly with the appropriate dummy objects to form the modified netlist.

4. The method of claim 1, wherein the step of replacing at least one object comprises the step of replacing each object in the list of unused objects with the appropriate dummy object to form the modified netlist.

5. The method of claim 1, wherein the step of forming the list of unused objects comprises the steps of parsing a file to extract a list containing object names for all used instances for the target architecture and parsing the netlist for the target architecture.

6. The method of claim 1, wherein the step of replacing comprises replacing the type of an instance for an object found in the repeated list of objects with a type for a corresponding dummy object if the object found in the repeated list is not on the list containing object names for all used instances.

7. The method of claim 1, wherein the method further comprises the step of parsing the netlist for the target architecture line by line and forming a modified netlist with the appropriate dummy objects when all lines of the netlist have been parsed.

8. The method of claim 7, wherein the method further comprises the step of feeding through a signal unchanged when simulating the appropriate dummy object during a simulation process using the modified netlist.

9. The method of claim 1, wherein the step of forming the list of unused objects comprises the step of manually composing a repeated listed of root objects specific to the target architecture.

10. The method of claim 1, wherein the step of replacing comprises the step of emptying a hardware description language version of a repeated object to form an object devoid of an explicit functional mapping of an input to an output.

11. A method of processing signals with a modified netlist within a software-based logic simulation tool comprising the steps of:

creating a list of repeated objects specific to a target architecture that are repeated objects;

emptying the repeated object found on the list of repeated objects forming a plurality of dummy objects;

parsing a file to extract a list containing object names for all used objects for the target architecture;

parsing a netlist of objects line by line for the target architecture;

replacing any object in the netlist with a corresponding dummy object from the plurality of dummy objects if the object in the netlist is not on the list containing object names for all used instances to form the modified netlist; and

simulating the modified netlist.

12. The method of claim 11, wherein the method further comprises the step of feeding through a signal unchanged when simulating the plurality of dummy objects during a simulation process using the modified netlist.

13. The method of claim 11, wherein the step of emptying the repeated objects comprises the step of emptying a hardware description language version of the repeated objects to form a plurality objects devoid of logic.

14. The method of claim 11, wherein the step of parsing the file comprises the step of parsing a file containing hierarchical path names to memory blocks of a field programmable gate array forming the target architecture.

15. The method of claim 14, wherein the method further comprises the step of generating the file containing the

hierarchical path names by converting bitstream names into Verilog hierarchical path names.

16. A machine readable storage, having stored thereon a computer program having a plurality of code sections executable by a machine for causing the machine to perform the steps of:

create a netlist of objects for the target architecture;

identify objects specific to the target architecture that are repeated regularly to identify potential dummy objects;

create a list of objects used by a design in the target architecture;

form a list of unused objects in the target architecture from the netlist of objects and the list of objects used by the design;

replace at least one object in the list of unused objects with an appropriate dummy object to form a modified netlist; and

simulate the modified netlist.

17. The machine readable storage of claim 16, wherein the computer program further has a plurality of code sections executable by the machine for causing the machine to perform the step of parsing the netlist for the target architecture line by line and forming a modified netlist with one or more of the appropriate dummy objects when all lines of the netlist has been parsed.

18. A system for simulating target architectures for implementation on field programmable gate arrays comprising:

means for creating a netlist of objects for the target architecture;

means for identifying objects specific to the target architecture that are repeated regularly to identify potential dummy objects;

means for creating a list of objects used by a design in the target architecture;

means for forming a list of unused objects in the target architecture from the netlist of objects and the list of objects used by the design;

means for replacing at least one object in the list of unused objects with an appropriate dummy object to form a modified netlist; and

means for simulating the modified netlist.

19. The system of claim 18, further comprising means for parsing the netlist for the target architecture line by line and forming a modified netlist with a plurality of dummy objects when all lines of the netlist has been parsed.

20. The system of claim 18, wherein the system uses a Verilog version of a hardware description language.